

**REMARKS**

Claims 1-36 are pending herein. In the Office Action, claims 1-4 and 20-23 were rejected under 35 U.S.C. §102(e) as being anticipated by Le et al. (“Le” USPN 6,158,047), and claims 5-19 and 24-36 were rejected under 35 U.S.C. §103(a) as being unpatentable over Le in view of Price (USPN 6,269,474).

Paragraph 38 of the Specification as filed is amended to replace the word “register” with “registry” as originally intended. In paragraphs 30, 31, 32, 38, 39 and 41 numerous references are made to the registry along with other examples of hardware specific information associated with a particular class or type of machine or equipment, such as the registry, drivers, configuration, information (INF) files, HAL, etc. (see, e.g., paragraph 38). At the amended location, the word “register” incorrectly appears instead of the intended word “registry” as evidenced by the other similar references. Applicant submits the use of the word “register” is a typographical error and requests approval of this amendment.

Prior to addressing the substantive rejections, a brief review of the disclosure as filed is provided. The present disclosure concerns a hardware agnostic manipulation and management of image resources for converting a disk drive image bootable by one server to a bootable disk drive image for another system with a different hardware configuration. As described in the present application, the operating system (OS) of a computer must be configured to operate for a given hardware computing platform capable of running the particular OS (paragraph [0002], or ¶2). Further, it is often desired to convert a bootable platform from one system to another (¶3). Stated another

way, it is desired to convert a disk drive or disk image configured for one system to another system with a significantly different hardware platform (§5).

In this context, the term “disk image” or “disk drive image” is often used to refer to the information on a physical disk (§4). A disk drive “image” is a portable description (e.g., a set of one or more files) of hardware persistent storage and includes hardware specific configuration information and drivers (§4). The present disclosure equally applies to physical systems as well as virtual systems. Virtualization software may be used to convert a single physical server into a pool of logical computing resources including one or more logical servers (a.k.a., virtual machines) (§4). For a virtual platform, the information contained on the bootable disk drive of the physical machine is converted to a disk drive image or image file, which may then mounted to a virtual platform to simulate a bootable virtual hard drive (VHD) for a logical server (§4).

In one embodiment, a conversion system performs an external introspection process (EIP) to convert the source image/disk into the converted target image/disk (§27). The conversion system generally performs the EIP to transform an image from one hardware platform to another or to convert between neutral and hardware-specific platforms (§27). The EIP enables conversion of a bootable disk drive compatible with a first platform (either physical or virtual) into a different bootable drive compatible with a different platform (either physical or virtual) regardless of the differences in the hardware configurations between the two platforms (§29). Disk image translation involves the translation of OS constructs and/or hardware specific information associated with a particular class or type of machine or equipment, such as the registry, drivers, configuration, information (INF) files, HAL, etc. (§30). In particular, disk image

translation involves system configuration modification (e.g., configuration data modification) and swapping out certain files (e.g., drivers and the like) to reflect a different hardware configuration and does not involve modifying any machine instructions or software code or the like.

The prior art references Le and Price show and describe techniques for translating or interpreting computer program code or low-level machine instructions and have nothing to do with converting a disk drive image bootable by one server to a bootable disk drive image for another system with a different hardware configuration. Le offers a way to run non-native code on a client machine that has an entirely different CPU architecture, without resorting to code interpreters. In Le, that which is being translated is *code* – the active logic involved in executing a computer program – rather than disk drive image translation. Likewise, Price concerns a *code* optimization system, whose purpose is the more efficient execution of a computer program rather than converting a disk drive image. Conversion of a disk drive image bootable by one computer system to a bootable disk drive image for another computer system with a different hardware configuration has very little if anything to do with code translation but instead enables an operating system image to be moved between different systems with different hardware configurations. In one embodiment, translation involves swapping out device drivers and modifying the operating system configuration, such as in the case of the registry of Windows®-based systems.

Applicant respectfully traverses the §102(c) rejection of claims 1-4 and 20-23 based on Le.

Le does not show “a first server that mounts the source disk image *as a* target disk drive” (emphasis added) as recited in claim 1. The cited portion of Le simply describes reading a source code module (SCM) 16 from a disk 14, providing optimized native code for the source file with a static translator using profile data 28, and storing the native code in a translation file 38. The cited portion of Le has nothing to do with mounting a disk image *as a* disk drive to a server.

Le also does not show “a repository that stores information and files useful for supporting the second hardware configuration” as recited in claim 1. The “second hardware configuration” is introduced in the preamble of claim 1, which recites a “conversion system for converting a source disk image supporting a first hardware configuration into a target disk image supporting a second and different hardware configuration.” The cited portion of Le refers to profile data 28, which “generally consists of one or more counters to count the number of time that each target is taken by a particular branch.” As described in Le, profiling generally means monitoring the behavior of a branch which terminates a source code block (Le, col. 5, lines 38-44). Applicant respectfully submits that profile data comprising counters for counting branches of computer code for computer code translation has nothing to do with a repository that stores information and files useful for supporting a second hardware configuration during conversion of a disk image for one hardware configuration into another disk image for the second hardware configuration.

Le also does not show “a rules library that facilitates conversion of hardware specific attributes in accordance with an external introspection process (EIP)” and “a conversion engine, executed on said first server and interfaced with said repository and

said rules library, that performs said EIP by examining the source disk image on said target disk drive to determine modifications to convert to the target disk image” as recited in claim 1. It is noted that the conversion engine performs the EIP by examining the source disk image of the target disk drive mounted by the first server. It is unclear how the cited portion of Le, which concerns a step of dynamically translating blocks of the SCM 16 to a native code, relates to the claimed rules library. Applicant certainly disagrees that the translation blocks or functions or files shown in FIG. 1 of Le illustrate the same disk image conversion process as claimed. As previously discussed, Le concerns *code* translation rather than disk drive image conversion as recited in claim 1.

Applicant respectfully submits, therefore, that claim 1 is allowable over Le. Claims 2-4 are allowable over Le as depending upon allowable claim 1. Applicant requests withdrawal of this rejection.

Further with respect to claim 2, the profile data 28 of Le comprises counters and the translation file 38 of Le is a storage file for the native code (Le, col. 5, lines 60-67) rather than a target profile retrieved from a repository used to determine modifications to convert a source disk image to a target disk image.

Further with respect to claim 3, it is noted that the profiler tool generates a target profile when executed on a target server having the second hardware configuration. As described in Le (col. 5, lines 31-38), the client executes the *source* program to profile the run-time behavior of the *source* block, which refers to the source rather than the target.

Further with respect to claim 4, the cited portion of Le (col. 4, lines 35-55) appears to have more to do with introducing the disclosed concept of translating a program into a native machine code rather than an inspector tool. In any event, Le does

not show an inspector tool that examines the source disk image of a source disk drive to generate a source profile as recited in claim 4.

Claim 20 is allowable over Le for similar reasons recited above with respect to claim 1.

Le does not show a method of converting a source disk image supporting a first hardware configuration into a target disk image supporting a second and different hardware configuration as recited in claim 20. Le does not show mounting the source disk image as a target disk drive on a first server as recited in claim 20. Le further does not show performing an external introspection process (EIP) by examining the source disk image on the target disk drive to determine conversion modifications to convert the source disk image to the target disk image as recited in claim 20. As noted above, Le concerns translation of code rather than conversion of disk images. Claims 21-24 are allowable for similar reasons recited above with respect to claim 2-4, respectively.

Applicant respectfully traverses the §103(a) rejections of claims 5-19 and 24-36 based on Le in view of Price.

Price concerns conversion of code (as also noted in the Office Action on page 5) and does not overcome the deficiencies of Le recited above with respect to claim 1 and 20. Price relates to a computer program optimization system which results in code reduction and standardization (Price, Field of the Invention or “Field”). Price also relates to a computer system which evaluates existing software and performs vertical, horizontal and sequential synchronization on the source code, utilizing an interim pseudo code to create new systems library and a program source code file (Price, Field). Price also concerns an optimization system including an option to convert existing code to a

codeless environment (Price, Abstract). Price has nothing to do with conversion of disk images as recited in claims 1 and 20.

Since claims 1 and 20 are allowable over Le and Price does not overcome the deficiencies of Le with respect to claims 1 and 20, claims 5-19 and 24-36 are therefore allowable over Le in view of Price as depending upon allowable claims 1 and 20, respectively. Applicant requests withdrawal of this rejection.

None of the amendments made herein were related to the statutory requirements of patentability, but instead were made for purposes of clarity and/or to remove extraneous and/or unnecessary language. Also, none of the amendments were made for the purpose of narrowing the scope of any claim.

**CONCLUSION**

Applicant respectfully submits that for the reasons recited above and for various other reasons, the objections and rejections have been overcome and should be withdrawn. Applicant respectfully submits therefore that the present application is in a condition for allowance and reconsideration is respectfully requested. Should this response be considered inadequate or non-responsive for any reason, or should the Examiner have any questions, comments or suggestions that would expedite the prosecution of the present case to allowance, Applicants' undersigned representative earnestly requests a telephone conference.

Respectfully submitted,

Date: August 28, 2007

By: /Gary Stanford/  
Gary R. Stanford  
Reg. No. 35,689

Gary R. Stanford  
Law Office of Gary R Stanford  
Customer Number 26122